

PENGEMBANGAN PERANGKAT LUNAK DETEKSI KECEPATAN KENDARAAN BERGERAK BERBASIS PENGOLAHAN CITRA DIGITAL

Farah Fajriyah dan Budi Setiyono

Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam

Institut Teknologi Sepuluh Nopember (ITS) Surabaya

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: budi@matematika.its.ac.id

Abstrak— Penelitian ini bertujuan untuk mendeteksi kecepatan kendaraan bergerak di suatu jalan dengan menggunakan pengolahan citra digital. Objek dari penelitian ini adalah video rekaman arus kendaraan. Dimana dalam perekaman video tersebut dilakukan beberapa pengukuran parameter untuk proses pendeteksian kecepatan. Parameter yang harus diukur diantaranya yaitu besar sudut pemasangan kamera, tinggi pemasangan kamera dan jarak horizontal kamera ke daerah titik awal yang tercakup oleh kamera. Kemudian dilakukan pemilihan area yang digunakan sebagai area pendeteksian kecepatan kendaraan bergerak. Selanjutnya yaitu proses *Background Subtraction* dengan metode *Gaussian Mixture Model*. Citra *foreground* difilter dan dilakukan proses morfologi untuk menghilangkan *noise* pada citra. Tahap berikutnya, dicari titik pusat dari kendaraan untuk mengetahui jarak. Selain itu juga dicari waktunya yaitu dengan mencari banyaknya *frame* saat kendaraan berada di dalam ROI. Dengan mengetahui jarak dan waktu, maka dapat dideteksi kecepatan dari kendaraannya.

Kata Kunci— Pengolahan citra digital, *Gaussian Mixture Model (GMM)*, *Speed Detection*.

I. PENDAHULUAN

Volume kepadatan arus lalu lintas setiap tahunnya mengalami peningkatan. Selain faktor kemudahan memiliki kendaraan bermotor, pertumbuhan penduduk yang cukup tinggi membuat kebutuhan akan transportasi juga semakin meningkat. Berdasarkan data dari Badan Pusat Statistik jumlah kendaraan bermotor pada tahun 2013 mencapai 104.118.969 unit[1]. Namun, pertumbuhan jumlah kendaraan tersebut tidak diimbangi dengan fasilitas jalan yang memadai, sehingga timbul permasalahan-permasalahan lalu lintas.

Salah satu diantaranya yaitu tidak seimbangnya jumlah kendaraan dengan ruas jalan. Hal ini mengakibatkan kondisi lalu lintas sehari-hari mengalami kepadatan. Hal ini diperparah dengan cara berkendara yang tidak sesuai dengan aturan, salah satunya adalah kebut-kebutan yang dilakukan oleh para pengguna kendaraan, sehingga rawan terjadi kecelakaan. Kecelakaan lalu lintas menjadi momok yang sangat menakutkan bagi para pengguna jalan raya. Bahkan kecelakaan lalu lintas menjadi salah satu penyebab kematian terbesar di Indonesia maupun secara global. Maka dari itu perlu dilakukan upaya untuk menertibkan perilaku berbahaya ini. Penentuan batas kecepatan maksimum yang ditunjukkan dengan rambu lalu lintas adalah salah satu aturan efektif untuk menanggulangi kebut-kebutan. Namun

tanpa ada petugas lalu lintas yang berjaga, pelanggaran kerap terjadi. Maka pengamatan kecepatan sangat penting untuk membantu pengawasan lalu lintas.

Dalam mengamati kecepatan kendaraan, kebanyakan masih menggunakan metode LIDAR (*Laser Infrared Detection and Ranging*) atau RADAR (*Radio Detection and Ranging*). Metode ini digunakan untuk mengurangi pelanggaran lalu lintas batas kecepatan kendaraan. Tetapi metode ini membutuhkan biaya yang besar untuk membeli alatnya serta membutuhkan keahlian khusus untuk mengoperasikan alatnya. Sebagai alternatif lain, saat ini pengawasan lalu lintas telah memanfaatkan kamera CCTV yaitu *road traffic monitoring center (RTMC)*. Sejalan dengan hal tersebut, saat ini pemerintah sedang membangun infrastruktur jalan, terbukti dengan pemasangan kamera CCTV pada perempatan jalan yang sudah banyak dilakukan. Hal ini dilakukan karena dianggap mampu melingkupi wilayah yang luas, sehingga diharapkan dapat meningkatkan keamanan. Namun pemasangan CCTV tersebut hanya dimanfaatkan sebagai pengawasan kepadatan lalu lintas saja, padahal video hasil dari rekaman kamera CCTV tersebut dapat juga digunakan untuk mendeteksi kecepatan kendaraan.

Beberapa penelitian yang terkait dengan Tugas Akhir ini diantaranya adalah penelitian yang dilakukan oleh Huei-Yung Lin, dkk dengan judul "*Vehicle Speed Detection from a Single Motion Blurred Image*"[2], kemudian penelitian yang dilakukan oleh Chomtip Pornpanomchai, dkk yang berjudul "*Vehicle Speed Detection System*"[3] dan penelitian yang dilakukan oleh Arash dkk, dengan judul penelitiannya "*Vehicle Speed Detection in Video Image Sequence using CVS Method*"[4]. Dari hasil penelitian tersebut, tingkat akurasi pendeteksian kecepatan kendaraan dipengaruhi dari intensitas cahaya yang diterima oleh video digital. Sehingga menyebabkan ketidakstabilan dalam pendeteksian kecepatan kendaraan.

Oleh karena itu, berdasarkan paparan latar belakang di atas, maka dalam Tugas Akhir ini akan dilakukan pendeteksian kecepatan kendaraan yang berbasis pengolahan citra digital dengan memanfaatkan video hasil rekaman kamera CCTV. Dalam penelitian ini penulis berencana membuat sistem perangkat lunak berbasis algoritma *Background Subtraction* untuk mendeteksi kecepatan kendaraan menggunakan metode *Gaussian Mixture Model* yang mampu beradaptasi dengan perubahan intensitas cahaya matahari.

II. DASAR TEORI

2.1 Algoritma Background Subtraction

Background Subtraction (BS) digunakan untuk mendapatkan objek yang bergerak pada serangkaian *image*. Objek yang bergerak dapat diidentifikasi dengan melakukan pengurangan antara *frame* pada waktu t dengan *background* model (latar belakang model). Kemudian nilai citra hasil pengurangan tersebut dibandingkan dengan nilai ambang batas (*threshold*) sesuai dengan metode yang dipakai untuk membangun *foreground image*[7].

2.2 Metode Gaussian Mixture Model

Gaussian Mixture Model (GMM) merupakan metode yang tepat untuk berbagai kondisi yang terdapat pada citra, seperti *background* citra yang selalu statis, multimodal, maupun yang mengandung *noise* (gangguan atau objek yang tidak diinginkan terdapat pada citra).

Model-model GMM terbentuk dari data warna piksel berdasarkan waktu. Hasil model tersebut akan menjadi 2 bagian, model yang mencerminkan *background* dan model *non-background*. Jumlah model GMM yang digunakan mempengaruhi jumlah model *background*[9].

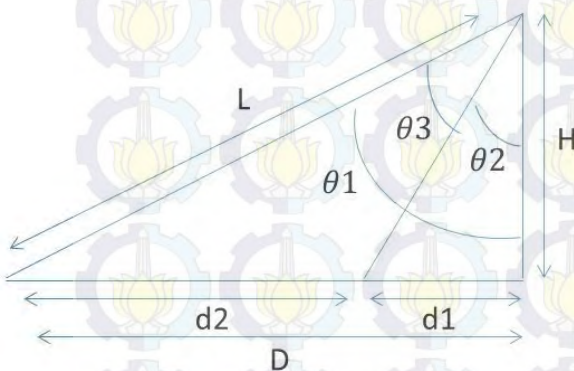
2.3 Metode Analisis Blob

Analisis *blob* ini menggunakan metode *connected component*, dimana di setiap kumpulan piksel yang tingkat keabuaannya bernilai satu, dikategorikan sebagai satu objek. Setiap objek yang terdeteksi akan diberi label untuk mempermudah pengolahan.

2.4 Tracking

Proses mencari obyek bergerak dalam urutan *frame* yang dikenal sebagai pelacakan. Pelacakan ini dapat dilakukan dengan menggunakan ekstraksi ciri benda dan mendeteksi obyek/benda bergerak di urutan *frame*. Dengan menggunakan nilai posisi obyek di setiap *frame*, bisa digunakan untuk menghitung posisi dan kecepatan obyek bergerak[8].

2.5 Paramter untuk Perhitungan Kecepatan Kendaraan dengan Posisi Kamera di Atas



Gambar 1. Setting Kamera dengan posisi di atas[4].

Berdasarkan Gambar 1, jika diasumsikan tidak ada kendaraan yang lewat, maka :

$$\theta_1 = \arctan\left(\frac{D}{H}\right) \quad (1)$$

dan sudut untuk daerah yang tidak terlihat oleh kamera adalah :

$$\theta_2 = \arctan\left(\frac{d_1}{H}\right) \quad (2)$$

dengan :

θ_1 adalah sudut pemasangan kamera;

θ_2 adalah sudut daerah yang tidak terlihat oleh kamera;

θ_3 adalah sudut pandang yang tercakup kamera;

H adalah tinggi kamera dari permukaan jalan;

D adalah jarak horizontal antara kamera dan kendaraan;

L adalah jarak nyata antara kamera dan kendaraan;

d_1 adalah daerah yang tidak terlihat oleh kamera;

d_2 adalah daerah yang terlihat oleh kamera;

Selain itu juga, dapat diketahui bahwa :

$$\theta_3 = \theta_1 - \theta_2 \rightarrow \theta_2 = \theta_3 - \theta_1 \quad (3)$$

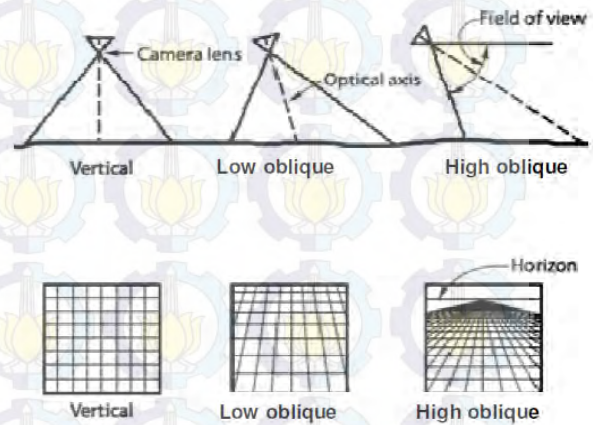
Daerah d_1 bisa diperoleh dengan :

$$d_1 = H \tan(\theta_1 - \theta_3) = H \tan \theta_2 \quad (4)$$

Dan untuk daerah d_2 bisa diperoleh dengan :

$$d_2 = D - d_1 \quad (5)$$

Kecenderungan dari kamera dengan posisi seperti Gambar 1 akan menghasilkan gambar dari tiga jenis area seperti yang ditunjukkan pada Gambar 2 yang bagian atas. Sedangkan Gambar 2 yang bagian bawah merupakan *grid* bagian garis yang terlihat pada berbagai jenis sudut kamera[4].



Gambar 2. Sudut kemiringan Kamera dan tampilan *grid* hasil rekaman dari atas[4].

2.8 Speed Detection

Kecepatan kendaraan diperoleh dari *frame* hasil deteksi *foreground*, yaitu dengan menentukan posisi kendaraan pada setiap *frame*. Jadi perlu ditentukan *boundingbox* dan *centroid* dari hasil yang telah dilakukan sebelumnya. *Centroid* disini sangat penting untuk mengetahui jarak kendaraan yang bergerak dalam *frame* yang berurutan.

Untuk mengetahui jarak yang ditempuh dalam pixel, dapat dihitung dengan mencari koordinat seperti berikut:

Titik koordinat objek saat berada pada *frame* $i = (x_1, y_1)$,
Titik koordinat objek saat berada pada *frame* $i + 1 = (x_2, y_2)$ [4].

Jarak perpindahan dari kendaraan yaitu:

$$d_1 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (6)$$

dan jika *framerate* dari video adalah 25 *frame* per detik, maka waktu antara dua *frame* berturut-turut yaitu 0,04s[4].

Dan untuk menghitung kecepatan dapat menggunakan persamaan berikut :

$$V = \frac{\Delta x}{\Delta t} \quad (7)$$

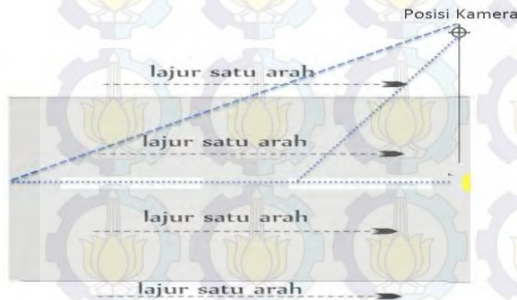
dimana Δx adalah jarak perpindahan dan Δt adalah waktu[4].

III. PERANCANGAN SISTEM

3.1 Perancangan Data

3.1.1 Data Masukan

Data masukan dalam sistem ini adalah video digital arus kendaraan dan beberapa parameter yang digunakan untuk proses pendeteksian, diantaranya yaitu : besar sudut pemasangan kamera, tinggi pemasangan kamera dan jarak horizontal kamera ke daerah titik awal yang tercakup oleh kamera. Data ini merupakan hasil pengambilan rekaman arus kendaraan di suatu jalan.



Gambar 3. Layout Untuk Pengambilan Video.

Tabel 1. Data Uji coba

No	Nama	Frame rate	Pixel
1.	Movie1.avi	25	1280x720
2.	Movie2.avi	25	1280x720
3.	Movie3.avi	25	1280x720
4.	Movie4.avi	25	1280x720
5.	Movie5.avi	25	1280x720
6.	Movie6.avi	25	1280x720
7.	Movie1.avi	30	1280x720
8.	Movie2.avi	30	1280x720
9.	Movie3.avi	30	1280x720
10.	Movie4.avi	30	1280x720
11.	Movie5.avi	30	1280x720
12.	Movie6.avi	30	1280x720

3.1.2 Perancangan Proses

Perancangan proses merupakan rancangan yang digunakan dalam proses pengolahan data masukan. Berikut ini beberapa perancangan prosesnya.

a. Mencari Titik Pusat dari Kendaraan

Proses mencari jarak perpindahan antar pixel dilakukan dengan mencari koordinat titik pusat dan menghitung jarak antar titik pusat. Untuk mendapatkan koordinat titik pusat maka terlebih dahulu mengolah objek-objek hasil dari analisis *blob*.

Secara sederhana, untuk menentukan titik pusat, langkah-langkah yang dilakukan adalah :

1. Menghitung koordinat x, dengan :

$$\text{Posisi } x = \text{bbox}(1) + \left(\frac{\text{bbox}(3)}{2} \right) \quad (8)$$

2. Menghitung koordinat y, dengan :

$$\text{Posisi } y = \text{bbox}(2) + \left(\frac{\text{bbox}(4)}{2} \right) \quad (9)$$

dimana :

$\text{bbox}(1)$ = nilai matriks *boundingbox* kolom ke 1

$\text{bbox}(2)$ = nilai matriks *boundingbox* kolom ke 2

$\text{bbox}(3)$ = nilai matriks *boundingbox* kolom ke 3

$\text{bbox}(4)$ = nilai matriks *boundingbox* kolom ke 4

b. Pengukuran Jarak

Untuk mengukur kecepatan sebuah obyek, harus diketahui informasi atas variabel yang berpengaruh, diantaranya variabel jarak dan variabel waktu. Variabel jarak didapat dari jarak diantara dua titik pusat obyek dari *frame* satu dengan lainnya yang bisa dihitung dengan menggunakan persamaan (6)[4].

c. Transformasi jarak ke jarak sebenarnya

Pengukuran jarak yang didapat dari tahap sebelumnya yaitu dalam satuan *pixel*, oleh karena itu jarak tersebut harus diubah ke jarak sebenarnya. Jarak kamera serta sudut pemasangan kamera untuk perekaman obyek sangat berpengaruh pada transformasi jarak. Untuk menghitung hubungan antar *pixel* dengan jarak sebenarnya adalah :

$$z = \frac{1}{x/y} \quad (10)$$

dimana :

x = jarak jangkauan sebenarnya

y = jarak jangkauan pada citra

z = hasil hubungan antar *pixel* dengan jarak sebenarnya.

d. Menghitung Waktu

Untuk mendapatkan variabel waktu maka akan dicari waktu yang dibutuhkan obyek dari mulai *start ROI* hingga *finish ROI*. *Frame rate per second* adalah banyaknya *frame* yang ditangkap dalam satu detik. Untuk menghitung waktu bisa dihitung dengan menggunakan rumus :

$$\Delta t = f \times w \quad (11)$$

dimana :

f = jumlah *frame* (*frame*)

w = waktu yang diperlukan satu *frame* ($\frac{\text{detik}}{\text{frame}}$)

Δt = waktu (*detik*)

e. Deteksi Kecepatan

Deteksi kecepatan kendaraan bergerak merupakan langkah akhir dari semua proses, dengan tujuan mendapatkan informasi berapa kecepatan kendaraan tersebut dalam satuan *km/jam*. Setelah didapatkan jarak (Δs) dan waktu (Δt) maka bisa dideteksi kecepatannya menggunakan persamaan (7)[4].

3.1.3 Data Proses

Data proses merupakan data yang digunakan dalam proses pengolahan data masukan. Data proses ini diperoleh dari hasil pengolahan data masukkan sesuai dengan tahapan algoritma dan metode yang telah disusun. Berikut tabel yang menjelaskan tahapan dari data proses.

Tabel 2. Data Proses

No	Tahapan	Input	Output
1.	Input Awal	Video	Frame Citra
2.	Pilih Area ROI	Frame Citra	Citra ROI
3.	Hitung parameter	Parameter	Parameter
4.	Proses GMM	Citra ROI	Citra <i>Foreground</i>
5.	Filtering	Citra <i>Foreground</i>	Citra <i>Foreground</i> yang telah difilter
6.	Deteksi <i>Blob</i>	Citra <i>Foreground</i> yang telah difilter	Area <i>blob</i> (<i>centroid, weight, height</i>)
7.	Hitung koordinat titik pusat kendaraan pada	Objek hasil deteksi <i>blob</i>	Koordinat x dan y

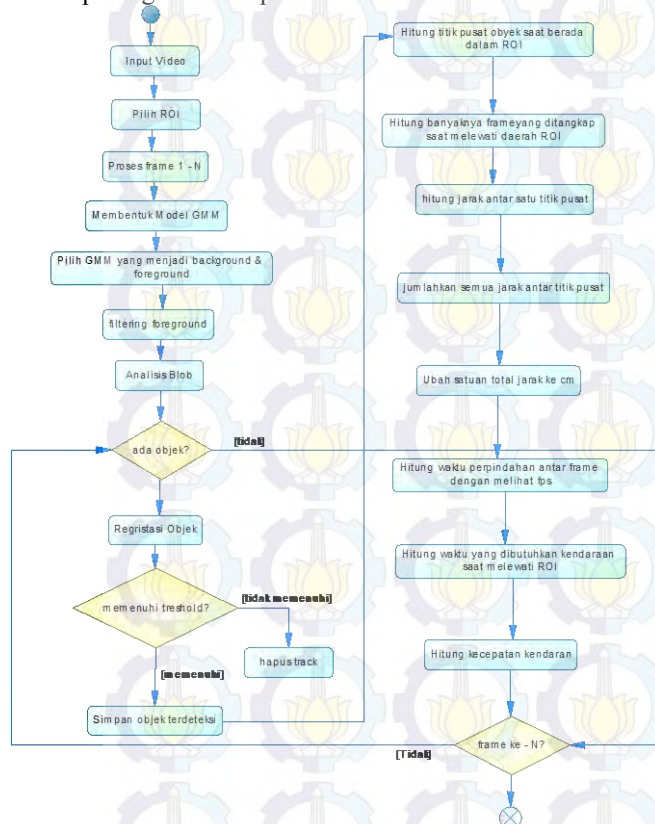
	saat melewati daerah ROI		
8.	Hitung jarak area yang dilewati kendaraan	Objek hasil deteksi <i>blob</i> dan parameter	Jarak
9.	Perhitungan waktu yang dibutuhkan	Objek hasil deteksi <i>blob</i> dan parameter	Waktu
10.	Hitung Kecepatan Kendaraan	Jarak dan Waktu	Kecepatan kendaraan

3.1.4 Data Luaran

Data keluaran berupa hasil deteksi kecepatan kendaraan bergerak.

3.2 Gambaran Sistem Secara Umum

Sistem perangkat lunak yang dibangun ini memiliki beberapa tahapan yang dijelaskan melalui diagram alir sistem perangkat lunak pada Gambar 4.



Gambar 4. Diagram alir sistem deteksi kecepatan kendaraan.

Berikut ini penjelasan dari setiap tahap dalam sistem perangkat lunak ini:

a. Akuisisi video

Data masukan berupa rekaman video digital *offline* arus kendaraan yang kemudian di-*scanning*. *Scanning* adalah proses pemecahan video menjadi beberapa rangkaian citra yang sering disebut dengan *frame*.

b. Penentuan ROI (*Region of Interest*)

Memilih ROI (*Region of Interest*) yang akan digunakan sebagai area penghitungan kendaraan bergerak. Pada penelitian ini digunakan ROI dengan luas area mencakup minimal setengah dari luas area *frame*.

c. Segmentasi citra

Kemudian sistem membagi *frame* menjadi citra *background* dan citra *foreground* dengan menggunakan algoritma *Background Subtraction* dengan metode *Gaussian Mixture Model*[11]. Citra *background* adalah citra yang menjadi latar belakang dari suatu objek yang

bergerak. Sedangkan *foreground* adalah gambaran objek bergerak yang terdeteksi.

d. Proses *filtering*

Citra *foreground* yang diperoleh di-*filter* untuk menghilangkan beberapa *noise* yang ada pada citra. Sehingga citra akan lebih mudah untuk diolah dan objek kendaraan akan tampak lebih jelas. Selanjutnya citra hasil *filtering* akan dilakukan penghilangan bayangan dan proses morfologi.

e. Deteksi objek kendaraan

Pendeteksian objek pada Citra hasil *filtering* dengan analisis *blob* untuk memberi label pada setiap objek yang terdeteksi. Analisis *blob* ini menggunakan metode *connected component*, dimana di setiap kumpulan objek yang tingkat keabuan pikselnya bernilai satu, dikategorikan sebagai satu objek. Setiap *blob* memiliki *centroid*, luas area, tinggi, dan lebar sebuah objek dari bentuk *rectangle*.

f. *Tracking* objek kendaraan

Langkah berikutnya adalah *tracking* objek kendaraan dengan *tracking blob*. *Tracking blob* ini untuk menentukan bahwa objek yang terdeteksi pada *frame* ke-*i* dengan *frame* ke-*i+1* adalah objek yang sama atau berbeda. Hal tersebut dapat diketahui dengan mengukur jarak dari *centroid* masing-masing *blob* yang terdeteksi. Jika jaraknya kurang dari *threshold* yang telah ditentukan maka objek yang terdeteksi pada *frame-frame* tersebut adalah objek yang sama.

g. Proses Deteksi kecepatan kendaraan

Objek hasil *tracking* tersebut akan dilakukan deteksi kecepatannya.



Gambar 5. Bagian (a) citra objek hasil *blob tracking* dan (b) merupakan hasil deteksi kecepatan.

IV. PENGUJIAN DAN PEMBAHASAN

Perangkat lunak ini dibangun menggunakan bahasa pemrograman atau compiler MATLAB 2013a

Selain itu juga diperlukan kamera digital untuk merekam arus kendaraan di jalan. Program ini dikembangkan menggunakan laptop dengan spesifikasi processor Intel Core i3-2310M CPU @2.10 GHz 4,0GB RAM, Intel HD Graphics 3000.

Uji coba pada program dalam Tugas Akhir ini dilakukan terhadap video *.avi. Pada proses pengujian akan dibandingkan hasil deteksi kecepatan dengan kecepatan kendaraan yang sebenarnya. Prosentase keberhasilan pendeteksian kecepatan kendaraan dihitung dengan persamaan berikut :

$$PK = \frac{KS - \text{abs}(KS - KP)}{KS} \times 100$$

dengan,

PK = Prosentase Keberhasilan,

KP = Kecepatan kendaraan yang terdeteksi program

KS = Kecepatan kendaraan sebenarnya

Pengujian program ini dilakukan dengan *input* video rekaman kendaraan bergerak di Halaman Sekolah. Pengambilan video dilakukan pada waktu sore hari yang

merepresentasikan kondisi memiliki bayangan. Durasi rekaman yang digunakan adalah 15 detik dengan objek yang direkam dalam video tersebut adalah motor, sepeda dan orang jalan kaki.

Pengujian dilakukan dengan memasukkan beberapa nilai parameter, diantaranya yaitu :

1. $D = 2400 \text{ cm}$
2. $H = 500 \text{ cm}$
3. $\theta_1 = 80^\circ$
4. $\theta_2 = 10^\circ$
5. $\text{Frame} =$ sesuai banyaknya frame dari hasil rekaman video yaitu 25 atau 30.

Setelah sebuah parameter itu dimasukkan selanjutnya akan dilakukan proses perhitungan untuk mendapatkan hasil skala perbandingan jarak nyata dengan jarak pada video. Kemudian dilakukan perhitungan kecepatannya yaitu dengan menjalankan hasil rekaman video pada program. Saat kendaraan berjalan dan kemudian memasuki daerah ROI maka pada saat kendaraan berada pada daerah ROI tersebut akan dilakukan proses perhitungan untuk mendapatkan kecepatan dari kendaraan tersebut. Proses perhitungan disini didapat dari proses deteksi objek kendaraan yang kemudian diproses juga dengan parameter yang sudah didapatkan pada proses sebelumnya. Setelah kendaraan keluar dari daerah ROI maka akan diperoleh kecepatan kendaraan saat berada pada daerah ROI tersebut.

Dengan menggunakan video rekaman sebanyak 3 kali percobaan dari masing-masing kendaraan menghasilkan perhitungan kecepatan kendaraan sebagai berikut :

Hasil deteksi kecepatan untuk masing-masing video disajikan pada tabel 3, 4, 5, 6, 7 dan 8.

Tabel 3. Tabel pendeteksian kecepatan kendaraan pada video Movie1.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Motor	25	18	19,52
Motor	30	18	19,18

Tabel 4. Tabel pendeteksian kecepatan kendaraan pada video Movie2.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Motor	25	20	21,49
Motor	30	20	21,17

Tabel 5. Tabel pendeteksian kecepatan kendaraan pada video Movie3.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Motor	25	24	24,70
Motor	30	24	25,23

Tabel 6. Tabel pendeteksian kecepatan kendaraan pada Movie4.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Jalan	25	7	7,798
Sepeda	25	11	10,38
Jalan	30	7	8,581
Sepeda	30	11	11,23

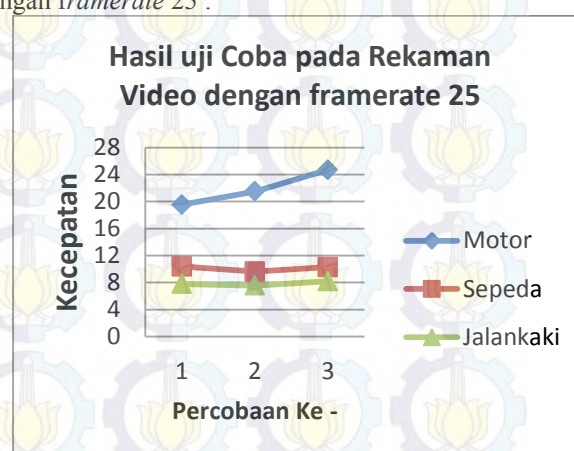
Tabel 7. Tabel pendeteksian kecepatan kendaraan pada video Movie5.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Jalan	25	7	7,561
Sepeda	25	10	9,629
Jalan	30	7	7,953
Sepeda	30	10	10,38

Tabel 8. Tabel pendeteksian kecepatan kendaraan pada Movie6.avi

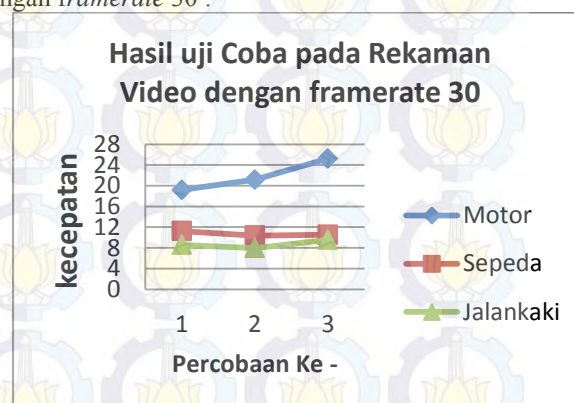
Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Jalan	25	8	8,198
Sepeda	25	11	10,29
Jalan	30	8	9,561
Sepeda	30	11	10,53

Berikut grafik dari pendeteksian kendaraan pada video dengan framerate 25 :



Gambar 6. Grafik pendeteksian kecepatan dengan framerate 25

Berikut grafik dari pendeteksian kendaraan pada video dengan framerate 30 :



Gambar 7. Grafik pendeteksian kecepatan dengan framerate 30

Maka, tingkat akurasi keberhasilan program dalam mendeteksi kendaraan bergerak dapat dilihat pada tabel berikut:

a. Rekaman Video Movie1.avi

Tabel 9. Tabel Prosentase Keberhasilan Video Movie1.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)	PK
Motor	25	18	19,52	91,56%
Motor	30	18	19,18	93,44%

b. Rekaman Video Movie2.avi

Tabel 10. Tabel Prosentase Keberhasilan Video Movie2.avi

Kenda raan	Framera te	Sebenarnya (km/jam)	Program (km/jam)	PK
Motor	25	20	21,49	86,89%
Motor	30	20	21,17	94,15%

c. Rekaman Video Movie3.avi

Tabel 11. Tabel Prosentase Keberhasilan Video Movie3.avi

Kenda raan	Framera te	Sebenarnya (km/jam)	Program (km/jam)	PK
Motor	25	24	24,70	97,08%
Motor	30	24	25,23	94,87%

d. Rekaman Video Movie4.avi

Tabel 12. Tabel Prosentase Keberhasilan Video Movie4.avi

Kenda raan	Framera te	Sebenarnya (km/jam)	Program (km/jam)	PK
Jalan	25	7	7,798	88,6%
Sepeda	25	11	10,38	94,45%
Jalan	30	7	8,581	77,41%
Sepeda	30	11	11,23	97,90%

e. Rekaman Video Movie5.avi

Tabel 13. Tabel Prosentase Keberhasilan Video Movie5.avi

Kenda raan	Framera te	Sebenarnya (km/jam)	Program (km/jam)	PK
Jalan	25	7	7,561	91,98%
Sepeda	25	10	9,629	96,29%
Jalan	30	7	7,953	86,38%
Sepeda	30	10	10,38	96,2%

f. Rekaman Video Movie6.avi

Tabel 14. Tabel Prosentase Keberhasilan Video Movie6.avi

Kenda raan	Framera te	Sebenarnya (km/jam)	Program (km/jam)	PK
Jalan	25	8	8,198	97,52%
Sepeda	25	11	10,29	93,54%
Jalan	30	8	9,561	80,48%
Sepeda	30	11	10,53	95,72%

Selanjutnya pengujian dengan menggunakan video rekaman arus kendaraan *real* di Jalan Raya dan di Jalan Tol dengan masukan parameter yang berbeda.

a. Rekaman Video DeltaMobil.avi

Tabel 15. Tabel Pendeteksian Kecepatan Kendaraan pada DeltaMobil.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Mobil1	25	-	42,78

b. Rekaman Video DeltaMobil2.avi

Tabel 16. Tabel Pendeteksian Kecepatan Kendaraan pada DeltaMobil2.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Mobil1	25	-	27,53
Mobil2	25	-	30,32

c. Rekaman Video jalantol99.avi

Tabel 17. Tabel Pendeteksian Kecepatan Kendaraan pada jalantol99.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Objek1	25	-	81,02
Objek2	25	-	83,68
Objek3	25	-	73,26
Objek4	25	-	57,80

d. Rekaman Video jalantol100.avi

Tabel 18. Tabel Pendeteksian Kecepatan Kendaraan pada jalantol100.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Objek1	25	-	95,84

e. Rekaman Video jalantol105a.avi

Tabel 19. Tabel Pendeteksian Kecepatan Kendaraan pada jalantol105a.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Objek1	25	-	92,31
Objek2	25	-	96,27

f. Rekaman Video jalantol105b.avi

Tabel 20. Tabel Pendeteksian Kecepatan Kendaraan pada jalantol105b.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Objek1	25	-	86,10
Objek2	25	-	48,99

g. Rekaman Video jalantol106a.avi

Tabel 21. Tabel Pendeteksian Kecepatan Kendaraan pada jalantol106a.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Objek1	25	-	91,42
Objek2	25	-	51,64
Objek3	25	-	53,33

h. Rekaman Video jalantol109b.avi

Tabel 22. Tabel Pendeteksian Kecepatan Kendaraan pada jalantol109b.avi

Kendaraan	Framerate	Sebenarnya (km/jam)	Program (km/jam)
Objek1	25	-	57,80
Objek2	25	-	78,73
Objek3	25	-	79,99

V. KESIMPULAN

Berdasarkan analisis terhadap hasil pengujian yang telah dilakukan terhadap sistem deteksi kecepatan kendaraan, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Tugas Akhir ini telah berhasil melakukan deteksi kecepatan kendaraan dengan tahapan: pemilihan ROI, segmentasi citra dengan metode GMM, proses filtering,

deteksi *blob* dan *tracking*, pendeteksian kecepatan kendaraan.

2. Untuk kecepatan objek yang paling cepat, tingkat akurasi keberhasilannya lebih baik jika menggunakan parameter *framerate* 25.
3. Dengan masukan parameter yang berbeda, tingkat akurasi keberhasilannya lebih baik jika menggunakan parameter tinggi sebesar 500cm.
4. Dengan menggunakan hasil dari percobaan yang terdapat data sebenarnya, hal tersebut bisa digunakan dasar untuk percobaan yang tidak diketahui data kecepatannya. Dan hasil percobaannya sudah cukup sesuai.
5. Tingkat akurasi keberhasilan yang lebih baik, lebih banyak muncul pada uji coba yang menggunakan parameter *framerate* 30.
6. Akurasi keberhasilan program dalam menghitung kendaraan bergerak dengan menggunakan parameter *framerate* 25 dan 30 yaitu prosentasi tertinggi 97,9% dan prosentase terendah adalah 77,41%
7. Perubahan cuaca dari cahaya matahari dan hembusan angin mempengaruhi performa program dalam mendeteksi kecepatan kendaraan.

DAFTAR PUSTAKA

- [1] bps.go.id. (2013). "Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis tahun 1987-2013". http://www.bps.go.id/tab_sub/view.php?tabel=1&id_subyek=17¬ab=12, posted: 2013. Diakses pada tanggal 16-02-2015
- [2] Lin, Huei-yung. Li, Kun-Jhih. Chang, Chia-Hong. (2008). "Vehicle Speed Detection from a Single Motion Blurred Image". **Image and Vision Computing** 26, Hal 1327-1337
- [3] Pompanomchai, C., and Kongkittisan, K. (2009). "Vehicle Speed Detection System". **IEEE International Conference on Signal and Image Processing Applications**, Hal. 135-139.
- [4] Rad, A. G, Deghani, A, Karim. M. R. (2010). "Vehicle Speed Detection in Video Image Sequence using CVS Method". **International Journal of the Physical Sciences** Vol. 5(17), Hal. 2555-2563
- [5] Rafael C. Gonzales, Richard E. Woods. (2002). "Digital Image Processing". **United States of America: Tom Robbins Publisher.**
- [6] Al Bovik. (2000). "Handbook of Image and Video Processing". **San Diego: Academic Press Publisher.**
- [7] Bharti, T. Tejinder. (2013). "Background Subtraction Techniques-Review". **International Journal of Innovative Technology and Exploring Engineering**, Vol. 2, Issue 3, Hal. 166 - 168.
- [8] Alper Yilmaz, Omar Javed and Mubarak Shah. "Object tracking: A survey". **ACM Comput. Surv.**, 38(4):13,2006.
- [9] Lazuardi, R. Arif Firdaus. (2014). "Penghitungan Kendaraan Bergerak Berbasis Algoritma Background Subtraction Menggunakan Metode Gaussian Mixture Model". **Tugas Akhir. Jurusan Matematika ITS.**
- [10] C. Stauffer, W.E.L. Grimson. (1999). "Adaptive Background Mixture Models for Real-time Tracking". **The Artificial Intelligence Laboratory, Cambridge.**
- [11] Hariyanto, Zamroji. (2015). "Klasifikasi Jenis Kendaraan Bergerak Berbasis *Geometric Invariant Moment*". **Tugas Akhir. Jurusan Matematika ITS**